IJTRET "Voice of Innovative Minds"

doi.org/10.54473/IJTRET.2024.8202

International Journal of Trendy Research in Engineering and Technology
Volume 8 Issue 2 April 2024
ISSN NO 2582-0958

STEERING ANGLE DETECTION IN AUTONOMOUS CARS USING DEEP LEARNING

Aishwarya N , Amogh P Gowda , K P Kiran , Thushar R , Sonali N M
Dept. Of Computer Engineering in (AI&ML)
Vidyavardhaka College of Engineering India, Karnataka, Mysore - 570 002

Received 08 April 2024 Received in revised form 15 April 2024 Accepted 18 April 2024
Available Online 25 April 2024

ABSTRACT

Steering angle prediction plays a crucial role in the control of Autonomous Vehicles (AVs) and has garnered the interest of researchers, manufacturers, and insurance companies within the automotive industry. Various Deep Learning (DL) architectures have been utilized to forecast the steering angle of AVs across different scenarios. An examination focusing on steering angle prediction through deep learning algorithms can assist seasoned researchers in pinpointing areas that necessitate further advancement. Additionally, newcomers in the field can utilize this study as a foundation. This paper offers an extensive analysis of the recent progress achieved in DL architectures concerning the steering angle prediction of AVs. A comprehensive taxonomy outlining the application of DL in steering angle prediction of AVs has been developed. The survey provides a succinct synthesis, summary, and analysis of research findings. It is evident that Convolutional Neural Network (CNN) is the preferred choice among researchers for predicting the steering angle of autonomous vehicles, compared to other DL architectures. Furthermore, existing research challenges have been identified. The primary obstacle encountered in DL-based steering angle prediction of AVs is the scarcity of real-world datasets, leading researchers to heavily rely on data generated from simulated environments. Lastly, potential alternative approaches to address the identified research challenges have been suggested, indicating promising avenues for future research endeavours.

Keywords Autonomous: Vehicles, Deel Learning, CNN.

I. INTRODUCTION

Human errors contribute to more than 90% of car accidents, with mechanical failures only accounting for 2%. Autonomous Vehicles (AVs) are being developed to address these errors, potentially saving thousands of lives each year. Research is focused on various aspects such as human recognition, traffic analysis, road and lane detection, steering mechanisms, and path planning. Currently, there are over 1,400 autonomous vehicles, including cars and trucks, being tested by more than 80 companies in 36 states across the United States. California is one of the states where AVs are already being tested on public roads. The progress in AV control has been significant in recent years, with many manufacturers aiming for commercial production within the next two to three years. Deep Learning (DL) has played a crucial role in enhancing AV control, especially in predicting steering angles by processing raw data without labels and focusing on key objects at different levels. Neural networks have the capability to understand complex relationships between features, making the task of steering a vehicle through traffic challenging to program using traditional algorithms. Various algorithms, including Genetic Algorithms (GA), fuzzy logic-based control learning, and steering autopilot control algorithms, have been utilized to achieve these objectives. However, the effectiveness of fuzzy logic-based controllers for preventing rear-end collisions depends on the number of fuzzy rules, which can impact their efficiency. This article offers a detailed overview, synthesis, and evaluation of recent advancements in steering angle prediction for AVs using DL methodologies.

II. LITERATURE REVIEW

A. Behavioural cloning for lateral motion control of autonomous vehicles using deep learning[1]

The autonomous vehicle sector has witnessed significant advancements in recent years, thanks to collaborations between car manufacturers and technology giants such as Google and Uber. These partnerships have led to the development of cutting-edge solutions for the next generation of autonomous vehicles. One notable example is NVIDIA's demonstration of an end-to-end learning model for autonomous lateral control. This model utilizes a

convolutional neural network to analyse road images and generate appropriate steering angles. To ensure ease of use and safety during training, the model is trained using a simulation environment like TORCS (the open racing car simulator). The training process involves acquiring data, with 70% of the dataset utilized for training, amounting to approximately four hours of data. The network architecture incorporates lambda layers for image normalization, 2D convolution layers, and fully connected layers. The research paper concludes that even with a limited amount of training data, the network successfully navigated the car within the simulation. An intriguing future enhancement could involve incorporating autonomous longitudinal control by incorporating vehicle speed and steering angles in the network's training process.

B. *Real-time self-driving car navigation using deep neural network[2]*

Autonomous vehicles represent a promising field of study in Artificial Intelligence, incorporating various technologies such as laser, radar, GPS, image processing, computer vision, and machine vision. Convolutional Neural Networks (CNNs) have emerged as the most widely used deep learning framework since their inception in the early 1990s, demonstrating cutting-edge performance in classification tasks. Researchers are currently exploring the use of DNN-based end-to-end control for cars and other robotic systems. This study introduces an independent car platform driven by AI technology, tackling a unique challenge in computer vision: enabling a car to autonomously navigate based solely on visual input from its camera. The model leverages a deep convolutional neural network to predict steering angles of the vehicle directly from real-time camera data. The architecture is trained using the Cross-Entropy Cost Function as the primary objective, with Adam serving as the optimizer. The trained model has shown a satisfactory level of accuracy, successfully guiding the car through various driving conditions on two different tracks.

C. *Self-driving car steering angle prediction based on deep neural network an example of CarND udacity simulator[3]*

Self-driving vehicles and other automated robotic systems are increasingly becoming a part of our daily lives, potentially saving numerous lives. However, the challenge of autonomous navigation in various environments necessitates vast amounts of data and computational power. Contemporary solutions rely on sensors, hardware, and software, requiring the creation of a real-time 3D model of the surroundings to analyse the situation and make decisions regarding movement. Recently, machine learning techniques have been utilized to steer self-driving cars without human intervention, utilizing extensive datasets. Various companies offer data to train neural networks for autonomous vehicles, such as Udacity, which supplies datasets and an emulator for this purpose. These datasets consist of images from three cameras (front, left, and right) captured during driving, along with the corresponding steering wheel angles. Neural networks are trained on these images to accurately predict steering angles. The aim of this study is to gather synthetic data and simulate the car movement process using Udacity's emulator through deep neural networks. Evaluating this approach will help determine its effectiveness in quickly collecting, modelling, training, and testing data. It is crucial to design neural network architectures with fewer parameters to predict steering angles based on the images obtained from self-driving cars. Researchers utilized the CarND Udacity Simulator, based on Unity, for their experiments, enabling them to simulate car movement in both manual and autonomous modes. They developed multiple convolutional neural network architectures, focusing on reducing the number of trained parameters. The findings revealed that while this architecture has significantly fewer parameters, it lacks an adequate number of convolutional layers for feature extraction compared to the initial architecture.

D. *Interpretable learning for self-driving cars by visualizing causal attention[4]*

Progress has been made in the development of self-driving vehicle control, as auto manufacturers are targeting commercialization within a 2-3 year timeframe. Visual attention networks offer spatial attention maps that are easily understandable for users. These networks filter out irrelevant image content and send the relevant information to the main recognition network for final decision-making. Instead of steering angle commands, the model predicts continuous steering angle commands from input raw pixels in an end-to-end manner using the inverse turning radius uˆt at each timestep t. In order to decrease computational expenses, each raw input image is down-sampled and resized to 80×160×3. A single exponential smoothing method is employed to mitigate the impact of human-related performance variations and measurement noise. The research is concentrated on creating a sophisticated self-driving vehicle controller utilizing a visual attention model enhanced with an additional layer of causal filtering. The model demonstrates competitive predictive

International Journal of Trendy Research in Engineering and Technology
Volume 8 Issue 2 April 2024
ISSN NO 2582-0958

capabilities and reduced complexity by eliminating features that do not significantly influence the output.

E. *End to end learning for self-driving cars[5]*

CNNs have transformed the field of pattern recognition by autonomously learning features from training examples. This method is especially effective in tasks related to image recognition, as it effectively captures the 2D characteristics of images while requiring minimal parameters to be learned. While CNNs have been utilized commercially for more than two decades, their popularity has surged recently due to advancements like the availability of large, labelled datasets such as the Large Scale Visual Recognition Challenge (ILSVRC) and the implementation of CNN learning algorithms on massively parallel GPUs. Data gathering involved driving on a variety of roads under different lighting and weather conditions. The network structure comprised 9 layers, which included a normalization layer, 5 convolutional layers, and 3 fully connected layers. The input image was divided into YUV planes and fed into the network. Training procedures encompassed frame selection, augmentation, simulation, and on-road testing. The evaluation of the networks was carried out in two stages: simulation and on-road tests. During simulation, the networks provided steering instructions for a set of pre-recorded test routes, while on-road tests measured the car's autonomous steering performance. The research illustrated that CNNs are capable of learning the entire task of lane and road tracking without the need for manual breakdown into road or lane marking detection, semantic abstraction, path planning, and control. Even a small amount of training data, less than a hundred hours of driving, was adequate to train the car to function effectively in various conditions, including highways, local roads, and residential areas under different weather conditions.

F. *End-to-end autonomous driving: An angle branched network approach[6]*

Recently, researchers have been investigating the application of imitation learning in driving end-to-end vehicles, utilizing image inputs similar to human beings. However, this approach falls short in determining whether a vehicle should turn left or right at an intersection. To tackle this issue, a method based on conditional imitation learning and navigation commands has been proposed. Driving along a given path involves considering the presence of other vehicles and pedestrians. The suggested network architecture, known as the angle branched network, incorporates sequential images, speed, and subgoal angle to calculate the current subgoal angle. The subgoal angle represents the angle between the ego-vehicle's heading direction (ht) and the subgoal direction (sdt). The network architecture comprises a VGG11 model pretrained on imagenet, convolution layers with 512 kernels, a global average pooling layer, two fully connected layers with 512 hidden units, and a fusion layer. Data collection is carried out using Carla's autopilot, and data pre-processing is crucial. The majority of collected data pertains to straight driving, causing the policy trained on unbalanced data to primarily focus on driving straight. To address this issue, sampling techniques are employed. The results demonstrate the effectiveness of the new navigation command (subgoal angle), which utilizes the vehicle's current position and current subgoal to calculate the subgoal angle. Two combinations of different images are used as observation inputs, namely rgbs and rgbds. The pixel values of each channel in the rgb image and the depth image are normalized to ensure a similar numerical scale. In conclusion, the proposed navigation command utilizes the vehicle's current position and current subgoal to calculate the subgoal angle in the end-to-end driving model. The model architecture adopts a branched structure and incorporates various types of images as observation inputs. It is found that incorporating depth information can reduce collisions with other vehicles and pedestrians, which is particularly significant in real-world scenarios.

G. *DeepPicar: A lowcost deep neural network-based autonomous car[7]*

Autonomous vehicles have sparked significant interest as a result of advancements in hardware and software technologies that enable fully autonomous driving. Deep neural networks (DNNs) have proven to be effective in various perception and control tasks for autonomous vehicles. Researchers are currently exploring DNN-based end-to-end real-time control for robotics applications. Nevertheless, there are numerous challenges associated with running these AI workloads on embedded computing platforms. A key objective is to replicate NVIDIA's DAVE-2 system on a smaller scale using the affordable multicore Raspberry Pi 3 platform. The Raspberry Pi 3 is equipped with a Broadcom BCM2837 SoC featuring a quad-core ARM Cortex-A53 cluster that can run at speeds of up to 1.2GHz. Achieving real-time control through inference timing necessitates a high control loop frequency to enable rapid responses to changing environmental conditions. Experiments were carried out to assess various real-time aspects of the DeepPicar, such as measuring worst-case

execution time (WCET), the impact of utilizing multiple cores to accelerate inferencing, coordinating multiple deep neural network models, memory bandwidth-intensive co-runners, shared L2 cache partitioning, and memory bandwidth throttling to ensure consistent real-time performance. The research revealed that all tested embedded platforms, including the Raspberry Pi 3, can support CNN-based real-time control ranging from 20 Hz to 100 Hz, depending on the specific platform. Furthermore, all platforms demonstrated the ability to integrate multiple CNN models and tasks. However, managing shared resource contention remains a critical consideration to guarantee the desired real-time performance on these shared memory-based embedded computing platforms.

H. *Model fusion: Weighted N-version programming for resilient autonomous vehicle steering control [8]*

This article presents initial findings on the utilization of redundancy to enhance the resilience of Deep Neural Network (DNN)-based algorithms for predicting steering angles in autonomous vehicles (AVs). The authors suggest an approach inspired by N-version programming (NVP), which involves combining outputs from various DNN models that are independently designed with different numbers of layers and parameters. They employ a weighted voting scheme to merge three community models developed by Udacity's self-driving car project, aiming to enhance the performance of each individual DNN. The preliminary results demonstrate two significant improvements. Firstly, the weighted model fusion achieves, on average, 40% higher accuracy in predicting steering angles compared to the individual algorithms. This improvement is measured using the root mean square error (RMSE) across various algorithms and types of perturbation. Secondly, the voting scheme ensures robustness by mitigating the failure of a single DNN model. The enhanced accuracy and reliability of the combined system indicate a promising opportunity to complement the current "sensor fusion" practice in AV technology with "model fusion." The authors utilize three independently designed DNN-based algorithms for steering angle prediction (Chauffeur, Autumn, and Rambo) from Udacity's challenge. They apply the weighted N-version programming approach to these DNN models to address the asymmetry in reliability among different community models and their continuous outputs (steering angles). In this scenario, the proposed NVP + weighted voting scheme outperforms the utilization of Eq. 2 alone. However, it is important to note that developing an NVP model can be costly, as it requires training individual DNN models with varying structures and parameters. Future research should focus on analysing the associated costs of implementing the proposed NVP model and evaluating the improvement in reliability compared to individual models. This analysis should consider different DNN structures, training datasets, and failure rates.

I. *Explaining how a deep neural network trained with end-to-end learning steers a car[9]*

PilotNet utilizes a convolutional neural network (CNN) to generate steering angles based on input images of the road ahead, serving as an end-to-end learning system for self-driving cars. The system was developed to eliminate the necessity of manual rule coding, opting instead for a learning approach through observation. While initial outcomes were promising, significant enhancements are essential for PilotNet to operate autonomously without human intervention. In order to comprehend the decision-making process of the system, a straightforward technique known as salient object detection has been devised to highlight the most crucial areas of an image that influence steering angles. This method identifies salient objects by pinpointing areas in the image that correspond to regions with the highest activations in the feature maps. The activations from higher level maps act as masks for lower-level activations through an algorithm that averages the feature map activations. The salient objects identified by this method are pivotal in determining steering decisions. PilotNet is trained to recognize features that are meaningful to humans while disregarding irrelevant structures in the camera images. This ability is acquired through data-driven learning without the necessity of manually crafted rules. In fact, PilotNet can identify subtle visual cues that are unrelated to driving tasks. In summary, the salient object detection method employed by PilotNet is a straightforward and efficient approach that enhances the system's steering capabilities. By recognizing subtle visual cues, PilotNet can make well-informed decisions regarding steering angles.

III. METHODOLOGY

A. FRAMEWORK

There are various machine learning (ML) frameworks and libraries available that support the use of GPU accelerators to enhance the learning process[10]. Some of these frameworks allow the utilization of optimized libraries such as CUDA (cuDNN) and OpenCL to boost performance. The key advantage of

multi-core accelerators lies in their highly parallel architecture, which enables GPUs to accelerate computations involving matrix-based operations. The field of software development in the ML community is characterized by its dynamic nature and diverse layers of abstraction. Advanced deep learning (DL) platforms are gaining popularity, many of which are open source. Keras is a high-level neural networks API written in Python that can run on top of Tensorflow, CNTK, or Theano. It has been widely adopted by researchers and industries in recent years. Caffe is a DL framework designed with emphasis on expression, modularity, and speed. Developed by Yangqing Jia at the Berkeley Artificial Intelligence Research (BAIR) and the BVLC community at UC Berkeley, Caffe offers an expressive architecture with GPU support, primarily for image classification since 2014.TensorFlow is an open-source software used for numerical computation involving data flow graphs. It serves as a comprehensive platform with a flexible ecosystem of tools, libraries, and community resources, enabling researchers and developers to deploy cutting-edge ML applications. TensorFlow's advantages include supporting DL research and development through its numerical library for data flow programming, outperforming other DL tools, efficiency in multi-GPU setups, and featuring a simple and adaptable architecture.

B. NETWORK

The Convolutional Neural Network (CNN) is a deep learning architecture that consists of multiple layers and is highly effective in solving image processing problems. It has been utilized in various applications, including the steering control of autonomous vehicles (AVs)[11]. Although the CNN was able to learn significant features of the road from a limited training signal, it exhibited poor robustness and visualization. The researchers also used the CNN to predict steering angles based on images captured ahead ofthe road. The results demonstrated that the CNN successfully identified objects and learned features such as lane markings, road edges, other cars on the road, bush lining, and typical vehicles.Another study referenced as[11] proposed the use of CNN for autonomous lateral control. The model generated appropriate steering angles, allowing the vehicle to complete laps without human intervention. However, the training data was limited and did not cover a wide range of scenarios. Two separate models were trained to predict the speed and steering angle, and the model with dual action performed better[12]. However, it exhibited slow camera latency performance.Lastly, reference [13] proposed a CNN-based closed-loop feedback system called DAVE2SKY to predict steering wheel angles for lateral control of AVs.13=20 The proposed CNN successfully controlled the steering wheel angle for lateral control of the autonomous vehicle and achieved robust control even in partially observable situations.

C. OPTIMIZER

In this segment, we introduce the solvers utilized for optimizing DL architectures in the training phase of various projects aimed at predicting the steering angle of AVs. The Adam solver stands out as the most popular choice, as evidenced by the literature review. There are several solvers accessible for updating weights and bias after each iteration. Researchers examine different solvers and assess their impact on the training process. Each model undergoes multiple training sessions with various solvers to determine the optimal optimizer for the final DL model. Commonly used solvers by researchers encompass the Adam solver, which is based on adaptive moment estimation, the stochastic gradient descent (SGD) solver, and Nesterov's accelerated gradient (NAG) solver. The Adam solver calculates adaptive learning rates for each parameter and is a gradient-based optimization approach. Reference Optimizers aid in determining the adaptive learning rates of neural networks. Furthermore, the selection of the solver is deemed a crucial aspect in the training of neural network . Adam is an effective technique for stochastic optimization that relies solely on first-order gradients and has minimal memory requirements.

IV. CONCLUSION

An overview of AV steering control using DL architectures is given in this article. The approaches that used DL for AV control are categorized, synthesized, and analyzed. Deep neural networks, convolutional neural networks, hybrid deep learning, and deep reinforcement learning are among the deep learning architectures in steering control that have been discussed in the literature review. The review revealed that the researchers' primary goal in using their approaches was to provide accurate steering angle forecasts. According to the review, researchers rely more on the convolutional neural network than any other DL design, indicating its significant importance in the field. There were identified research problems and potential solutions for resolving the challenges. Both novice researchers interested in DL steering control of unmanned aerial vehicles (AVs) and AV steering control experts can utilize our evaluation as a starting point and

benchmark for developing novel steering control algorithms.

REFERENCES

[1]. S. Sharma, G. Tewolde, and J. Kwon, ‘‘Behavioral cloning for lateral motion control of autonomous vehicles using deep learning,’’ in Proc. IEEE Int. Conf. Electro/Inf. Technol. (EIT), May 2018, pp. 228–233.

[2]. T.-D. Do, M.-T. Duong, Q.-V. Dang, and M.-H. Le, ‘‘Real-time self-driving car navigation using deep neural network,’’ in Proc. 4th Int. Conf. Green Technol. Sustain. Develop. (GTSD), Nov. 2018, pp. 7–12.

[3]. M. V. Smolyakov, A. I. Frolov, V. N. Volkov, and I. V. Stelmashchuk, ‘‘Self-driving car steering angle prediction based on deep neural network an example of CarND udacity simulator,’’ in Proc. IEEE 12th Int. Conf. Appl. Inf. Commun. Technol. (AICT), Oct. 2018, pp. 1–5.

[4]. J. Kim and J. Canny, ‘‘Interpretable learning for self-driving cars by visualizing causal attention,’’ in Proc. IEEE Int. Conf. Comput. Vis. (ICCV), Oct. 2017, pp. 2942–2950.

[5]. M. Bojarski, D. Del Testa, D. Dworakowski, B. Firner, B. Flepp, P. Goyal, L. D. Jackel, M. Monfort, U. Muller, J. Zhang, X. Zhang, J. Zhao, and K. Zieba, ‘‘End to end learning for self-driving cars,’’ 2016, arXiv:1604.07316. [Online]. Available: http://arxiv.org/abs/1604.07316V

[6]. Q. Wang, L. Chen, B. Tian, W. Tian, L. Li, and D. Cao, ‘‘End-to-end autonomous driving: An angle branched network approach,’’ IEEE Trans. Veh. Technol., vol. 68, no. 12, pp. 11599–11610, Dec. 2019.

[7] . M. G. Bechtel, E. Mcellhiney, M. Kim, and H. Yun, ‘‘DeepPicar: A lowcost deep neural network-based autonomous car,’’ in Proc. IEEE 24th Int. Conf. Embedded Real-Time Comput. Syst. Appl. (RTCSA), Aug. 2018, pp. 11–21.

[8]. A. Wu, A. H. M. Rubaiyat, C. Anton, and H. Alemzadeh, ‘‘Model fusion: Weighted N-version programming for resilient autonomous vehicle steering control,’’ in Proc. IEEE Int. Symp. Softw. Rel. Eng. Workshops (ISSREW), Oct. 2018, pp. 144–145.

[9]. M. Bojarski, P. Yeres, A. Choromanska, K. Choromanski, B. Firner, L. Jackel, and U. Muller, ‘‘Explaining how a deep neural network trained with end-to-end learning steers a car,’’ 2017, arXiv:1704.07911. [Online]. Available: http://arxiv.org/abs/1704.07911

[10]. Machine learning framework : Streamlining ML Model development, https://www.snowflake.com/guides/machine-learning-frameworks/

[11]. S. Sharma, G. Tewolde and J. Kwon, "Behavioral Cloning for Lateral Motion Control of Autonomous Vehicles Using Deep Learning," *2018* IEEE International Conference on Electro/Information Technology (EIT), Rochester, MI, USA, 2018, pp. 0228-0233,

[12]. Pedro J. Navarro 1,*, Leanne Miller 1, Francisca Rosique 1, Carlos Fernández-Isla 1 and Alberto Gila-Navarro, ‘End-to-End Deep Neural Network Architectures for Speed and Steering Wheel Angle Prediction in Autonomous Driving’Electronics 2021, 10, 1266.

[13]. Junekyo Jhung, Il Bae,Jaeyoung Moon,,Taewoo Kim,Jincheol Kim,Shiho Kim,‘End-to-End Steering Controller with CNN-based Closed-loop Feedback for Autonomous Vehicles’, Conference: 2018 IEEE Intelligent Vehicles Symposium (IV)